### # 项目概述

该项目是分类任务，由于数据集的性质，采用度量学习的方式来进行训练分类。

原项目路径:https://kevinmusgrave.github.io/pytorch-metric-learning/inference\_models/

参考项目路径:

https://github.com/xxcheng0708/pytorch-metric-learning-template/tree/main/utils

# 数据介绍

数据格式是xlsx，有4个表，分别是训练集，留出集，西溪验证集，浙一验证集

每个表的列数相同，其中第一列为类别(因变量)，其余类为自变量

其中训练集有318条数据，留出集有45条数据，西溪验证集有49条数据，浙一验证集有52条数据

在实验中还使用了total数据集(四个数据集的总和)和totalwithouttrain数据集(三个数据集的总和，去掉训练集)，计算相应的指标

### # 实验介绍

##### ## 模型架构介绍

模型架构1，trunk是MLP，embedding是MLP，classifier是MLP

其中trunk输入是32(不算batchsize)，隐藏层有3层，分别是64，64，64，输出是32

其中embedding输入是32，输出是64

其中classifier输入是64，输出是2

模型架构2，trunk是one CONV，embedding是MLP，classifier是MLP

其中trunk输入是1\*32的一维向量(不算batchsize)，有三层，第一层16次卷积，第二层32次卷积，第三层1次卷积，输出是1\*32

其中embedding输入是32，输出是64

其中classifier输入是64，输出是2

##### ## 超参介绍

batchsize为32，max\_epochs为200，三个组件的优化器都为Adam，三个优化器的lr都为0.0001，weight\_decay为0.0001，使用的是loss是两个loss的组合，TripletMarginLoss(margin=0.1)和CrossEntropyLoss()，TripletMarginLoss和CrossEntropyLoss权重比例是1比0.5，epsilon为0.1。

##### ## loss详细介绍

TripletMarginLoss：

输入的数据先经过embedding，变成长度为64的数据。

之后把这个数据输入到mine组件中，计算余弦相似度，得到batchsize\*batchsize大小的相似度矩阵。

之后用labels数据，用自己跟自己匹配,如果自己是(1,0,0)，则每个数据跟自己(1,0,0)进行匹配，得到(1,0,0)(0,1,1)(0,1,1)，并返回匹配矩阵和不匹配矩阵,之后分出两个矩阵，分别是正类矩阵和负类矩阵，都是继承相似度矩阵，在正类矩阵中把负类的值设置为很小的值，在负类矩阵中把正类设置为很大的值，之后进行升序排序，这样的话正类矩阵中前面的都是小的，负类矩阵中后面的都是大的 ，之后利用epsilon为0.1进行正类筛选困难数据和负类筛选困难数据(说明，即正类到目标的距离+epsilon小于负类到目标的距离)。

得到了正类负类原始的三元组数据之后，用相似度距离矩阵得到正类与原始目标的距离和负类与原始目标的距离，利用公式，正类距离-父类距离加epsilon小于0，使用的是loss是relu，即小于的设置为0，大于0的保持原样，这样就能区分正类到原始数据的距离和负类到原始数据的距离。

可参考这个视频

https://www.bilibili.com/video/BV1yz4y1R7dH/?spm\_id\_from=333.337.search-card.all.click&vd\_source=ef589c64d362730d47a50a4754b7fa82

CrossEntropyLoss，将经过trunk，embedding，classifier得到的二分类数据跟label数据进行计算loss

指标详细介绍

auc数据经过trunk，embedding，classifier得到的是两个数据，可以代表两个类别的概率,之后进行计算auc

acc数据经过trunk，embedding，classifier得到的是两个数据，可以代表两个类别的概率，之后进行计算auc

将embedding数据先使用knn,输入验证集，进行分类，每个数据集都有多个labels，用真实标签跟多个labels进行比对，之后记录每个数据的累计正确标签数量，并计算每个类别的精度，之后k-1，即用前k-1个label计算精度，之后计算每个数据的平均精度之后计算所有数据的平均精度，这个即为mean\_average\_precision 指标

precision\_at\_1是knn的k为1的特殊情况，即knn分类只有一个标签

R-Precision 在knn k不为1，但是只计算前R个knn的label的精度，r可以不等于k

还进行了交叉验证，即将训练集分成5份，选取其中一份为验证集，其余四份为训练集，之后换一份为验证集，分别进行5次实验，之后求平均的auc和acc

注:

所有实验都是已最优分类auc为运行指标，即模型以最优的auc进行保存,其余指标都是从最优的auc对应的epoch进行读取，且在相同epoch的情况下记录最小epoch的auc，如下表选择的是epoch101

|  |  |
| --- | --- |
| Epoch | Auc |
| 100 | 0.7 |
| 101 | 0.8 |
| 102 | 0.8 |
| 103 | 0.7 |

使用的训练集为对数归一化后的数据（十七支队）.xlsx的训练集

验证集为对数归一化后的数据（十七支队）.xlsx的训练集的西溪验证集